Surabhi Gandhi ***Contents:***

a) Requirements
b) Hardware Design
c) Software Design
d) Hardware and Software Testing
e) Validation

Requirements

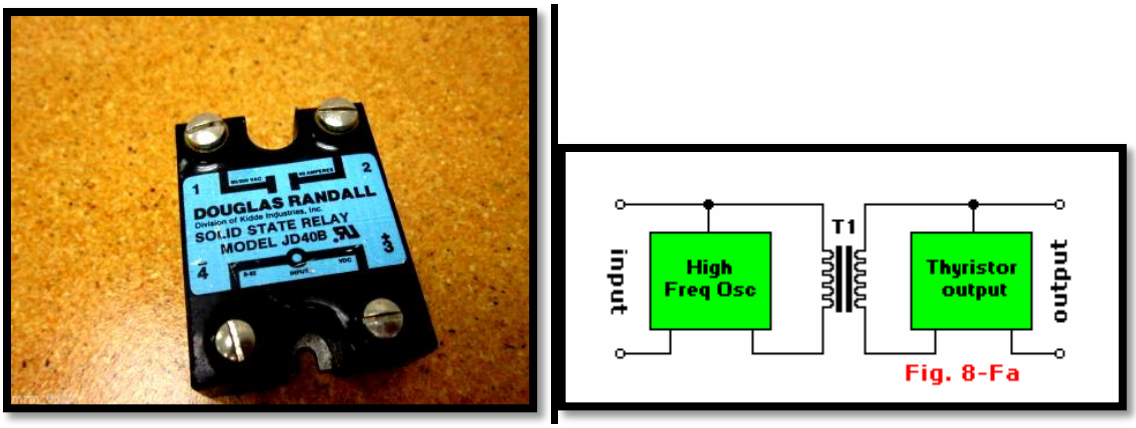
Design an AC Fan speed controller circuit with the following system requirements:

- System should have at least one binary input
- System should have at least one binary output,
- System must incorporate at least one transistor
- The C code must include at least two different interrupts
- There are 5 speeds.
- **Full speed,**
- **3% of duty cycle,**
- **97% of duty cycle,**
- **stop and**
- **10% of duty cycle (default).**

- The speed increases/decreases by ~1.5% when 3% or 97% duty cycle is selected creating multiple intermittent speeds of the fan.
- The transistor must act in Saturation; and be able to provide 3*20mA=60 mA current for the LED network.
- The LEDs individually must consume 20mA of current. Each LED must have a forward voltage of 1.7V capable of 1500 mcd typically
- Speed of fan should work on Pulse Width Modulation (PWM)
- The fan should be 120V AC consuming 0.8A of Current
- Fan should be within the range of the relay which is 300V and 40 Amps
- Specification of the relay:
  - ✓ DOUGLAS RANDALL SOLID STATE RELAY MODEL: JD40B
  - ✓ 300V AC 40 AMPERES

Hardware Design and Analysis

- The relay looks like:



Solid State Relay external and Internal

- Analysis of relay in comparison with the TRIAC CIRCUIT I was working on earlier:

Basics:

Solid state relay (SSR) means switching without any mechanical means. It is nothing more than a electronic switch, which can be a TRIAC, SCR OR Opt isolator. Virtually all SSR are Single Pole Single Throw, Normally Open (SPST=NO) devices; Where the outputs turn on in response to a control voltage. They take operating power from the main I/O though some may require DC Logic power.

The resistor is usually sized for a 5-volt logic input, and results in a specified "ON" range of 3-6 volts DC. For wider operating ranges (typically 4-32 volts DC), the resistor is replaced by a constant-current diode. Then, AC input circuits rectify and filter the control input before applying it to the LED.

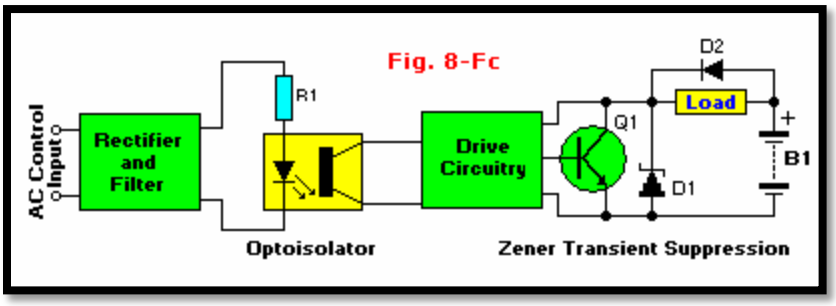
AC to DC:

The optoisolater photo current is amplified and used to drive whatever output device the relay is connected to. The power for the drive circuitry is taken from the output load or is supplied in separate. The following table summarizes it:

Table 1

Output	AC or DC	Package Style	Max Load Currents	Max Load Voltages	Volt Drop (on) at Rated Load	Leakage Current (off) At 25°C
Bipolar Transistor	DC	Dual Inline (DIP)	500mA 50mA	60Vdc 250Vdc	1 - 1.5V	20µA
		Input/ Output Module	3.5A 1A	60Vdc 200Vdc	1.2 - 1.75V	10µA - 1mA
		Power	2 to 20A	50Vdc	1.5 - 2V	5 - 15mA
		Power (High Voltage)	to 5A	250Vdc	2V	10µA
Thyristor (Triac or SCR)	AC	Dual Inline (DIP)	0.3 to 1A RMS (to 3A RMS on heat sink)	140 or 280V RMS	1.5V max	10µA - 1mA
		Input/ Output Module	3.5A RMS	140 or 280V RMS	1.5V max	2 - 5mA
		Power	10 - 40A RMS	140 to 280V RMS	1.5V max	2 - 15mA
Power MOSFET	AC DC	Dual Inline (DIP)	100 - 500mA (Some to 1A)	60 - 300V	Resistance 0.25 - 50 ohms	Resistance typically 100Meg

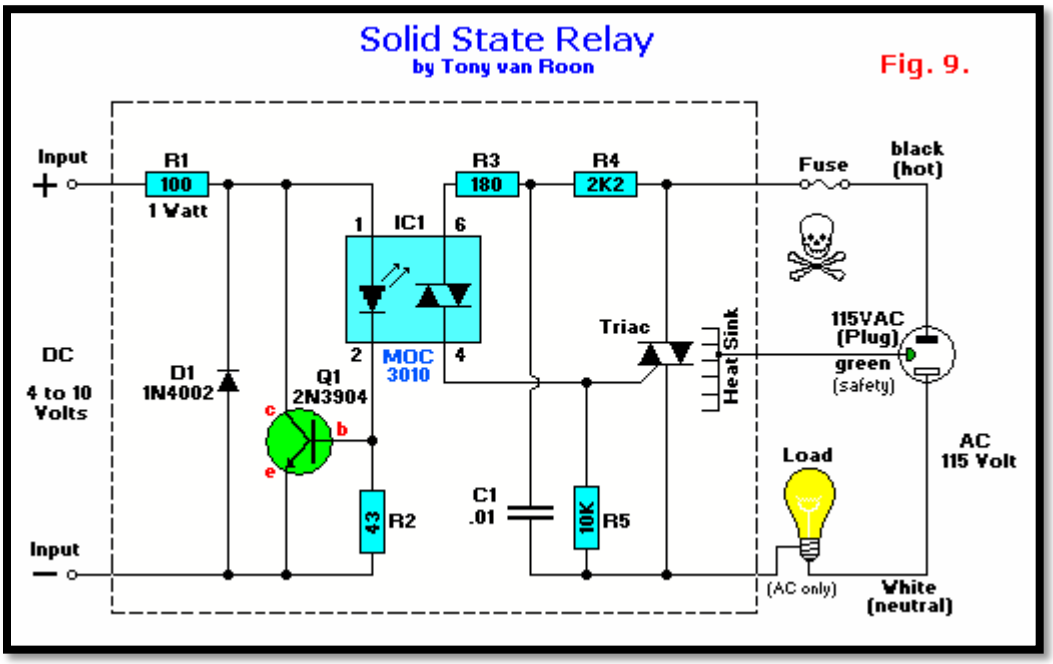
Also I/O isolation is used in Relays. Since I was working with AC mains I decided not to stick with the Triac as even the MOC 3051 optocoupler chip blew off and didn't provide enough isolation to me.
The solid-state relays use opto-isolators. All of at least 1.5 kilo-volt RMS I/O breakdown. Most of them have been rated, listed, or approved by safety agencies like UL, CSA, and VDE. Transformer coupling is also used to isolate solid-state relays.
The following figure explains it pretty well:



Zero Switching:

In normal operation the trigger side of the relay is totally asynchronous to the AC side. That means that a trigger could occur during any part of the sine wave. If the trigger occurs near a peak (90 or 270 degrees), a large current will flow into the load almost instantly. That creates a lot of RFI (Radio Frequency Interference) and also is very hard on the filament of ordinary light bulbs. In order to prevent that, zero-crossing SSR's accept the trigger at any time but delay turning on the AC load until the next time the AC voltage passes through zero volts.

Solid State Relay Internal Circuit:



For a simple SSR, an optoisolator such as the Motorola MOC3010 or the NTE3047 will be sufficient. For a zero-crossing SSR, a MOC3031 or NTE3049 will do. May companies make optoisolators. Make sure yours has a "Triac output" and that the pinouts are compatible with your design. Table 2 shows some typical Triac output optoisolator specifications. The models shown are already a decade old but still widely used and available.

Why NOT TRIAC?

TRIAC:

1) It is tough to find a TRIAC which can handle high voltage. The minimum for a 115 Volt AC line rewuires 200 Volt TRIAC. a 220 Volt line requires a 400 Volt TRIAC.

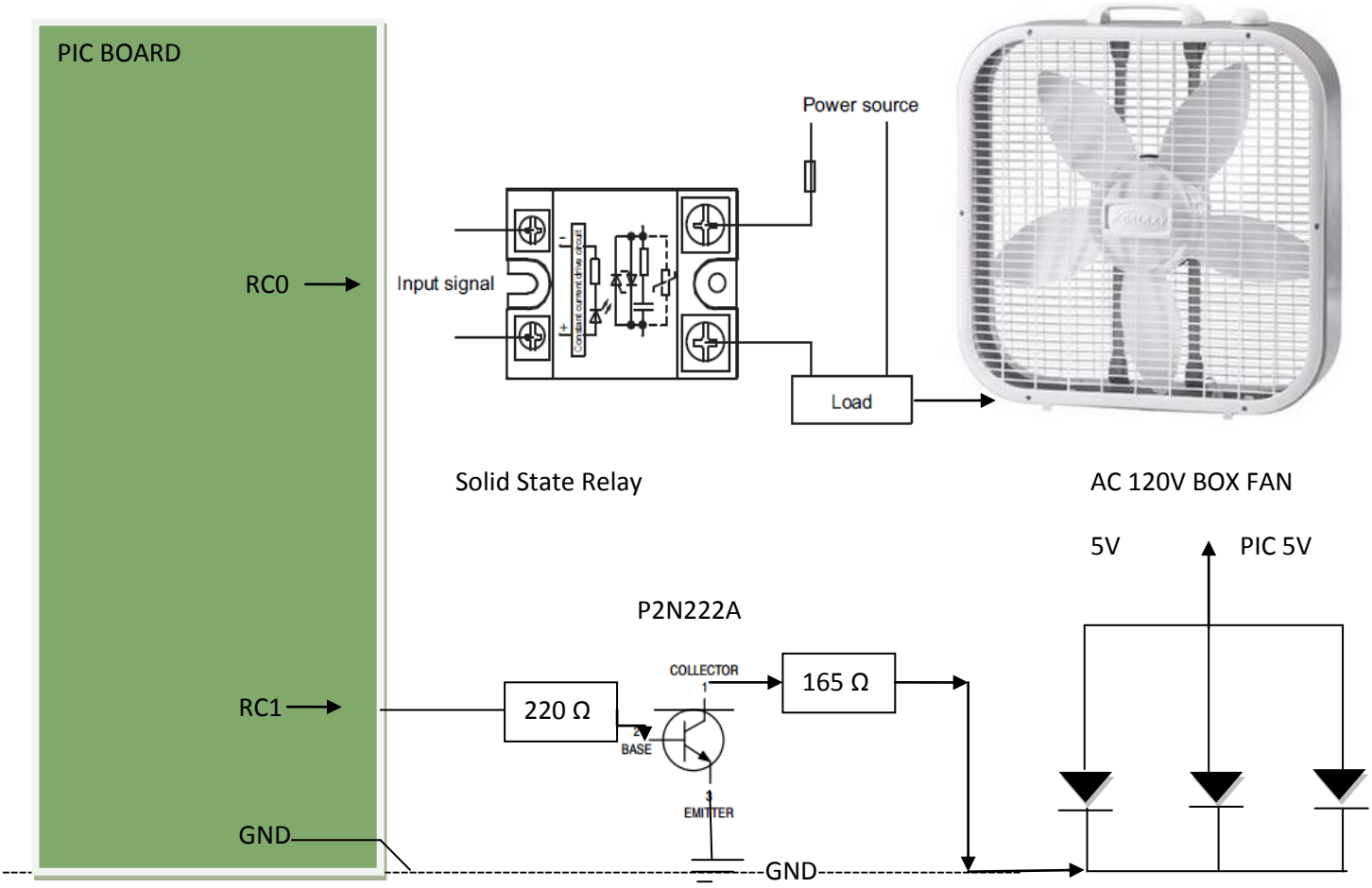
2) A 6 amps TRIAC will handle 6 amps only if it is properly heat slinked. Since motors draw a lot more current during startup this is not a good thing to work with.

3) It is tough to know the exact gate current for motor operations. For instance the MOC 3010 optosolator can provide 100mA of current we might need more in our operations.

Building an SSR requires putting 115/120 V AC on PCB which is a fairly fine idea however it is better to cover all PCB tracks on the 115/120 V AC side with a silicon sealer

The only precaution, other than the one about working with 115/220/220 AC volts, is to heatsink the Triac. If we leave the leads on the Triac long, it should be simple matter to find some heat sink to attach to the Triac. SSR's can only switch an AC line. Trying to switch a DC line will result in a relay that closes but never opens

Connection Diagram for the PIC-FAN Circuit



When the transistor is turned on there will be about a 0.7 V drop across the base emitter junction. Which means:

$$\frac{(5-0.7)}{20mA} = Rbase = 215\Omega$$

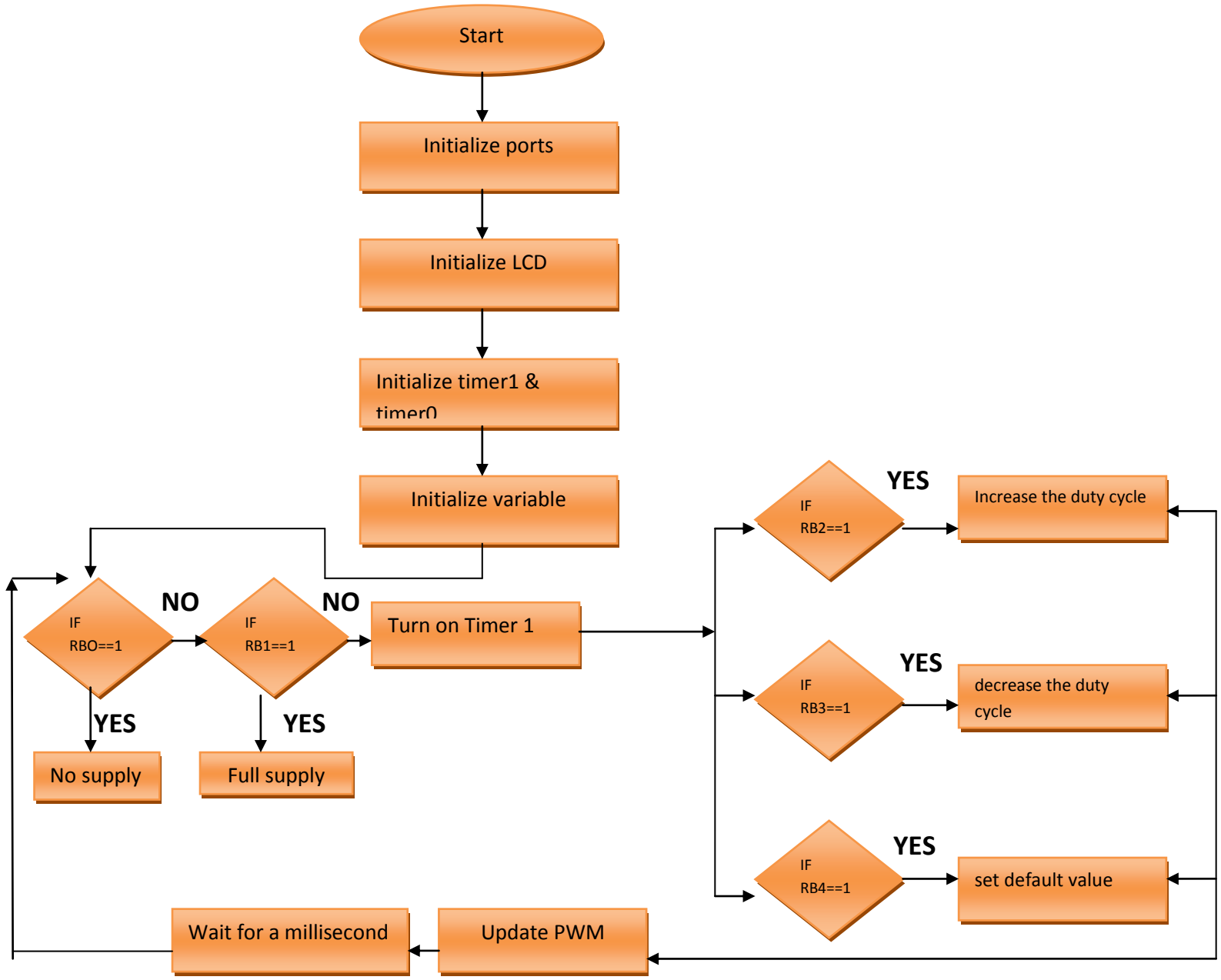
Please refer to hardware and software testing for calculations and analysis of this circuit.

Software Design:

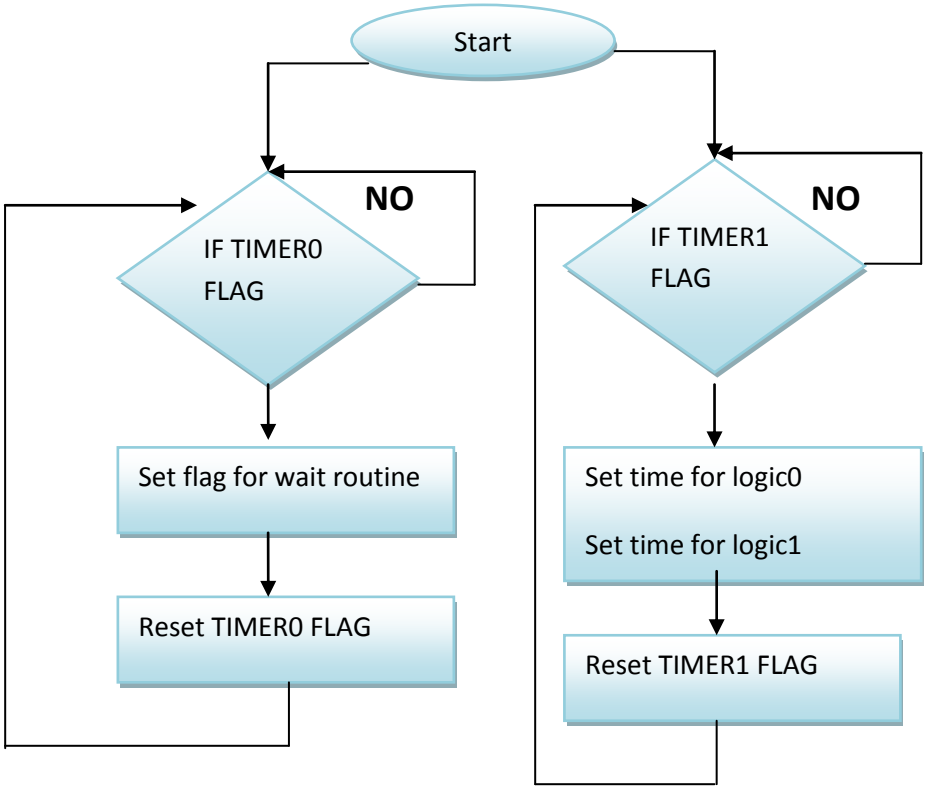
The C code is attached at the end.

Flow chart is as follows:

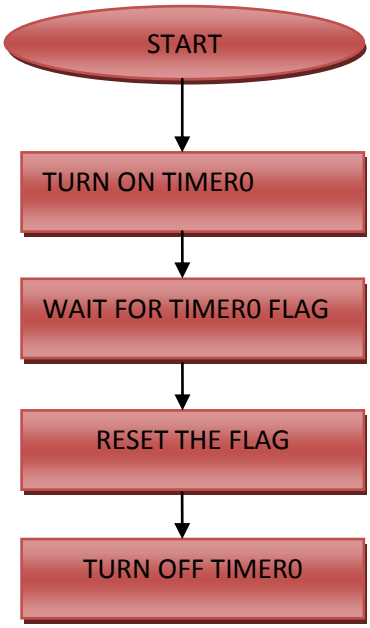
Main Loop Flowchart:



Interrupt routine:



Wait routine:



The software executes fast and is able to perform all functions as indicated by the flowchart.

Hardware and Software testing

ON CHARACTERISTICS

DC Current Gain	h_{FE}			–
(I_C = 0.1 mAdc, V_{CE} = 10 Vdc)		35	–	
(I_C = 1.0 mAdc, V_{CE} = 10 Vdc)		50	–	
(I_C = 10 mAdc, V_{CE} = 10 Vdc)		75	–	
(I_C = 10 mAdc, V_{CE} = 10 Vdc, T_A = –55°C)		35	–	
(I_C = 150 mAdc, V_{CE} = 10 Vdc) (Note 1)		100	300	
(I_C = 150 mAdc, V_{CE} = 1.0 Vdc) (Note 1)		50	–	
(I_C = 500 mAdc, V_{CE} = 10 Vdc) (Note 1)		40	–	
Collector – Emitter Saturation Voltage (Note 1)	$V_{CE(sat)}$			Vdc
(I_C = 150 mAdc, I_B = 15 mAdc)		–	0.3	
(I_C = 500 mAdc, I_B = 50 mAdc)		–	1.0	
Base – Emitter Saturation Voltage (Note 1)	$V_{BE(sat)}$			Vdc
(I_C = 150 mAdc, I_B = 15 mAdc)		0.6	1.2	
(I_C = 500 mAdc, I_B = 50 mAdc)		–	2.0	

For the P2N222A Transistor $V_{CE(SAT)}$ is 0.3 V when Ic=15-mA and Ib=15mA. This is something to note for further calculations and validation purposes.

In normal operation the value of Beta is at least 100, so we will be using that value. (Note that the Hfe value goes uptil 300)

We know from the circuit diagram explained before, The value of Rbase was calculted as: $\frac{(5-0.7)}{20mA} = Rbase = 215\Omega$

This value isn't critical so we use the closest standard value (220Ω). When the input to the base resistor is zero (or less than about 0.65V) no base current flows and the transistor is off (i.e. no current flows through the 220Ω resistor). When the input to the base resistor is 5V the base current is 20mA from the PIC and turns on the transistor.

The Red Led that we are using here, are rated as follows:

Part #	Color	Typical Vf @ 20mA	Typical mcd @ 20mA	Wavelength (nm)	Price ea (qty = 100)
MV-8104	Red 5mm LED	1.7V	1500 mcd	660nm	$0.21

- Vf is the turn-on voltage for this LED. If the first diode is on, the ideal diode model would have a1.7V drop across the diode.
- Typical mcd refers to the efficiency of the diode (mcd is a unit of light intensity).
- Wavelength is the average wavelength of the light from the LED (a more precise definition of the color than the term 'red').

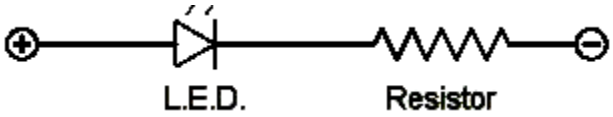
Note: Due to the NPN transistor the current becomes 20mA*beta (beta=100 or maximum 300) which can go up to 2-6A. Also, + of LED will get the supply from PIC we only need to ground this circuit.

Ic=βIb
so, Ic= 100*20mA, = 2A.

The transistor uses the cut off and saturation modes. The saturation current is amplified in order to provide each of the three LEDs in parallel a supply of 20mA.

Our LED specifies a forward current of 20mA at a forward voltage of 1.7 volts. We will be wiring the LED from a power source, and using a resistor to limit both the current and the voltage.

The voltage source is 5 Volts DC. Since the voltage across both device (Resistor and Led) is 5 Volts, and the LED is rated for 1.7 volts, the voltage drop across the series resistor must be 3.3 volts.



Since the current through two devices in series is the same through both devices, and we want the maximum current through the LED to be 20 mA, the resistor must drop 3.3 volts at a current of 20mA.

We need to calculate the Resistance of the resistor.

The law says that V=IxR. This can be re-arranged to say I = V/R or R = V/I. This 3rd form is the one we want:

Resistance = Voltage / Current
The math:

- R = 3.3 Volts / 20 milliAmps
- R = 3.3 Volts / .020 Amps
- R = 3.3/.020
- R = 165 Ohms.

So, we need:

- LED, forward current: 20mA, forward voltage: 1.7V
- Power supply: 5VDC
- Resistor, 165 Ohms

The next step is to choose a resistor with a value of at least 165 ohms. Due to manufacturing processes, resistors are manufactured with 5, 10, or 20% tolerance. The tolerance means the value may vary + or - 5% or 10% or 20% from the stated value. With a manufacturing tolerance of 10%, a 165 ohm resistor would limit the current to 3.3/181, or 18mA. This is below the 20mA maximum for the LED.

Limiting the current to less than 20 mA or reducing the voltage will extend the life of the LED, just as running a light bulb at less than rated voltage will extend its life! And, just as in a light bulb, reducing the current or voltage will reduce the brightness of the LED.

When wiring multiple LEDs in a circuit, it is best to connect them in parallel, each with its own series resistor to limit the current. In this case, the power source must be able to supply enough current in total to insure each device gets enough current. Three LEDs at 20mA each would need a total current of 60mA which we have discussed earlier

When the circuit is completely set up I took readings in the lab according to which:

1) On pressing RB0 the fan ideally turns off and the voltage across the transistor is 0.71V

2) On pressing RB1 the fan is set to full speed and the voltage across the transistor is 3.7V

The voltage on collector of NPN transistor is 4.3V (5V – 0.7V).

1) On pressing RB0 the fan slows turns off and the voltage at RC1 is 0V. RC1 is connected to the base of the transistor. V_{BE} = 0V which puts the transistor in cut-off region as E ≥ B < C. This renders the '-' terminal of LED floating and it turns off.

2) On pressing RB1 the fan slowly speeds to full. The voltage at the base (RC1) is now 5V and E < B > C which puts the transistor in saturation region and turns the LED on.

- Function of the following buttons on the PIC in conjunction with the LEDs
  - RB0 : Stop the Fan completely : LEDs must turn off
  - RB1 : Full speed of fan : LEDs must be fully bright
  - RB2 : Increase the speed according to the duty cycle : LEDs must go brighter with speed
  - RB3 : Decrease the speed according to the duty cycle : LEDs must go dimmer with speed
  - RB4: Default state: LEDs should be dimly on.

TIMER 0 AND TIMER 1 are being implemented in this code. TIMER 1 caters for the duty cycle increase and decrease; whereas TIMER0 does the wait routine to 1 millisecond.

TIMER0 calculation:

T0CS = 0 uses 20ns oscillator

T0CON = 0x88 or PS = 1

TMR0 = -5000 or Y = 5000

N = PS*Y = 5000

Time = 20ns * 5000 = 1ms

TIMER1 calculation:

TMR1CS = 0 uses 20ns oscillator

T1CON = 0xB1 or PS = 8

TMR1 = -5208 or Y = 5208

N = PS*Y = 41664

Time = 20ns * 41664 = 8.3ms

The wave toggles every 8.3ms which is a period of 16.6ms or 60Hz wave

For 10% duty cycle:

TMR1 switches between -9416 and -1000 which translates to :N = 8*9416 = 75328

Time = 20ns * 75328 = 15.66ms

N = 8*1000 = 8000

Time = 20ns*8000 = 1.6ms

Time fan is on is 1.6 ms ~ 10% duty cycle and rest of the time it is off

ISSUES AND SOLUTIONS WITH HARDWARE SOFTWARE INTEGRATION

Debouncing: PIC reaction time which can also be noted as the instruction time is microseconds since it is built up of semiconductors. Say for instance I press a button for 1 complete second; due to the processing speed being much faster than the manual action itself the PIC assumes we pressed the button 4-5 times in that span, which we originally just pressed one time. This is because the PIC is faster and processes much faster. So accordingly the PIC changed the duty cycle. Also, a circuit continually loops and runs the program until powered off. So, the board is continually reading (many times a second) the state of the switch. As you're starting to see, this bouncing can cause the board to believe many shots have been "requested" within a very narrow window of time.

I dealt with this issue by providing a wait statement. Now the code samples after every 1 millisecond. When I press it for more than one millisecond it will take it as 2-3 shots. So to match the processing time the duty cycle is ~1.5% which increases after each quick press step by step.

For example: When I press RB3 it slows the speed of the fan and that is because the duty cycle goes down by 1.5% at each press until eat reaches 3% of the total duty cycle.

- Pulse Width Modulation Duty Cycle and Oscope readings.
    - RB0 : duty cycle is 0 %hence there is no waveform output on the Oscope.
    - RB1 : Full speed of fan : duty cycle is 100 % and the current as displayed is 4.89 A.
    - RB2 : Increase the speed : has a broader pulse as expected with increasing duty cycle at steps of 1.5%.
    - RB3 : Decrease the speed : has a thinner pulse as expected with decreasing duty cycle at steps of 1.5%.
    - RB4: Default state: with a duty cycle of 10%.The output of the Box fan is as follows:120v AC 60Hz.

Validation:

I faced various issues with this project until it's completion. First major issue was finding the right TRIAC and insulation circuit. When that Circuit did not work as expected I used a Solid State relay which has a TRIAC internally built into it plus proper insulation from the main line. It is not a mechanical device so there is no chance of delay in processing times.

The final system consists of the PIC Microcontroller, SSR, AC box fan, and a breadboarded circuit with a parallel LED network indicating the increment and decrement in the speed of the fan by consuming more and less current respectively from the P2N222A Transistor.

This project meets all the requirements as stated in the Project guidelines. Namely:

- The hardware incorporates atleast one binary input and at least one binary output.
- We are using a P2N222A transistor
- The software uses two interrupts, TIMR1 AND TIMR0.

Knowledge of embedded systems is demonstrated by using resistors, transistors, LEDs and PIC microcontroller together with an AC load. All the hardware and the software is a result of carefully developed thought process. All the calculations are supported with Lab Results and the software justifies when the LED network runs in cohesion with the Fan speed.

The fan does incorporate 5 main features, full speed, stop, default, increase and decrease.

The transistor is in Saturation and also amplifies the base current such that it can power up all the three LEDs together. The saturation condition is satisfied

P2N2222A

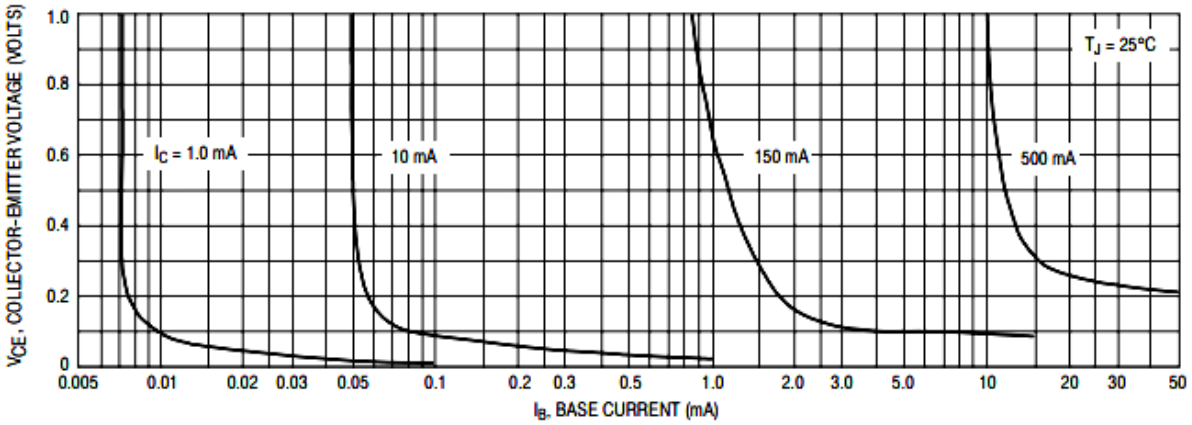


Figure 4. Collector Saturation Region

As we can see for 20mA base current the Vce is roughly 0.3V. We noticed earlier that our transistor was at 0.71 Volts which is well suiting the range. This justifies that the transistor is in the Saturation region. Also we know that gain of a transistor is related by the formula: $I_c=\beta I_{B.}$. Now the transistor is providing enough current for all the three LEDs to light up. The LEDs , each consumes 18mA of current hence taking care we dont fry them.

The pulses shoot at 10% and increase by 1.5%. The routine is set up in the code.

The full speed corrsponds to 100% duty cycle, the second set of speed corresponds to 3% of duty cycle, the third set corresponds to 97% duty cycle, and stop is 0% duty cycle. The default function renders to 10% duty cycle. The speed increases and decreases by a value of 1.5 % when 3% or 97% duty cycle is selected creating multiple intermittent speeds of the fanThe AC Signal verification is: